International Journal of Electronics,
Communication & Instrumentation Engineering
Research and Development (IJECIERD)
ISSN(P): 2249-684X; ISSN(E): 2249-7951
Vol. 4, Issue 2, Apr 2014, 39-46

TRANS STELLAR
• Journal Publications • Research Consultancy

AN OPTIMIZE TECHNIQUE FOR CHANNEL SECURITY

ESHANT G. RAJGURE[1] & AJAY P. THAKARE[2]

[1]Department of Electronics & Telecommunication, Sipna C.O.E.T, Amravati, Maharashtra, India

[2]HOD, Department of Electronics & Telecommunication, Sipna C.O.E.T, Amravati, Maharashtra, India

ABSTRACT

UART (Universal Asynchronous Receiver Transmitter) is a serial communication protocol; mostly used for long-distance, high speed, low-cost data exchange between computer and peripherals. During the actual industrial production, sometimes we do not need the full functionality of UART, but simply integrate its core part. UART includes three basic modules which are the baud rate generator, receiver and transmitter. The UART implemented with VHDL language can be integrated into the FPGA (Field Programmable Gate Array) to achieve compact, stable and reliable data transmission. It is also significant for the design of SOC.

In this project Paper we are concentrating on one of the most secured way of serial communication by automatic generation and detection of Baud Rate. To achieve auto bauding we adopt configuration of UART using FPGA. UART controller is designed based on FPGA that provide low cost, high performance logic solutions for applications having complex control systems and meet their secured communication demands quickly and efficiently.

During Communication Unwanted Receiver gets try to intrude common data on channel, In that case it has raw data but they cannot identify the original form of data transmitted. It is because of continuous variation of baud rate by baud generator is detected by uart (receiver) but not by microcontroller (intruder). Original form of data is different than what it collects. This system is reconfigurable and scalable and it is used to reduce the synchronization error between the subsystems with in a system. After studying this project paper we definitely proved that this is "The Optimum Technique for Secured Communication".

KEYWORDS: UART, VHDL, VLSI, FPGA

INTRODUCTION

A universal asynchronous receive/transmit (UART) is an integrated circuit which plays the most important role in serial communication. It handles the conversion between serial and parallel data. Serial communication reduces the distortion of a signal, therefore makes data transfer between two systems separated in great distance possible. Asynchronous serial communication has advantages of less transmission line, high reliability, and long transmission distance, therefore is widely used in data exchange between computer and peripherals. Asynchronous serial communication is usually implemented by Universal Asynchronous Receiver Transmitter (UART). UART allows full-duplex communication in serial link, thus has been widely used in the data communications and control system. This project paper uses VHDL to implement the UART core functions and integrate them into a FPGA chip to achieve compact, stable and reliable data transmission.

Communications between the master controller (PC) and slaver controllers (μc) are implemented by serial or parallel port. Parallel communication needs a lot of multi-bit address bus and data bus and it is only convenient for short distance transmission where as Serial communication is another way of communication used extensively because of its simple structure and long transmission distance. In some complex control systems or in controllers and processors such as Digital Signal Processors (DSP) the communication demands would affect the performance of the entire system. It is difficult to attain the desired result for various factors affecting the systems in terms of communication. Nevertheless, sometimes a common serial port could not meet requirements of complex systems with different Baud Rates equipments. Even with some special Baud Rate equipments, it is not possible to meet the desired requirement. The communication parameters such as Baud rate, which is the measure of transmission speed in asynchronous communication, Bit Error Rate (BER), and the synchronization between sub-systems, also engender great effect. Therefore, the device performance will be affected by these parameters. In addition, to make good use of control algorithms within the complex systems and to improve their precision, it is desirable to use UART, because it is not possible to implement this multi-baud rate communication system without a special baud rate converter.

If During Communication attacker takes data on channel, In that case it has raw data but they cannot identify the original form of data transmitted. It is because of continuous variation of baud rate by baud generator is detected by uart (receiver) but not by microcontroller's (attacker) present on the channel. Original form of data is different than what it Collect; in this way we have best Channel Security by Using UART communication technique.

LITERATURE REVIEW

Dr. Garima Bandhawarkar Wakhle, Iti Aggarwal, and Shweta Gaba, Electronics and Communication Department ASET, Amity University Noida, India present paper on 'Synthesis and Implementation of UART using VHDL Codes' in IEEE, 2012 International Symposium on Computer, Consumer and Control. The proposed paper describes the universal asynchronous receiver/transmitter i.e. UART which is the kind of serial communication protocol which allows the full duplex communication in serial link. This paper presents the hardware implementation of a high speed and efficient UART using FPGA. The UART consists of three main components namely transmitter, receiver and baud rate generator which is nothing but the frequency divider. This has been simulated on Model Sim SE 10.0a and has been implemented by using Verilog description language which has been synthesized on FPGA kits such as Virtex4 and Spartan3. This uses Verilog description language to get the modules of UART. After studying the comparative analysis we conclude that there is a difference in between the number of slices, LUTs, GCLKs and the maximum frequency. The results are quiet stable and reliable and has great flexibility with high integration. If we use FIFO in making the UART our design becomes more flexible, stable and reliable which provides the high bps rate.

Jan Henning Mueller, Mojdeh Hamzavi Nejad Moghaddam, Bastian Mohr, Sebastian Strache, Ralf Wunderlich and Stefan Heinen Chair of Integrated Analog Circuits and RF Systems RWTH Aachen University, Aachen, Germany published paper under title 'An Adaptable UART Based Configuration and Read-out Interface for IC Prototypes'. This paper presents an easy-to-use UART (universal asynchronous receiver transmitter) based communication pro- tool named SERIAS which can configure a test or prototype IC (integrated circuit) directly with a PC using the serial interface (EIA-232, former RS-232/V.10). For PCs without a serial interface, USB can be used, combined with an appropriate bridge IC. It is also possible to read data from the IC. The universal protocol is implemented in a hardware description language

(HDL) on the IC side and in MATLAB on the PC side and can be easily adapted to the requirements and the structure of the specific IC. The implementation can also be used to communicate with a prototype design on a field-programmable gate array (FPGA). A prototype IC configuration and read-out protocol SERIAS and its implementation in VHDL and MATLAB has been presented. It is easily adaptable and allows to use a PC with MATLAB to communicate directly with the IC. The system has been tested successfully using an FPGA.

System Architecture

Basic UART communication needs only two signal lines (RXD, TXD) to complete full-duplex data communication. TXD is the transmit side, the output of UART; RXD is the receiver, the input of UART. UART's basic features are: There are two states in the signal line, using logic 1 (high) and logic 0 (low) to distinguish respectively. It also supports configurable baud rate generator with data length of 8 bits per frame.

Frame Format

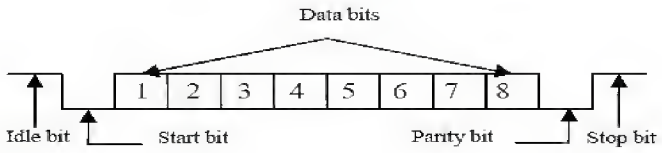

Figure 1

The UART frame format is shown above, UART's basic features are: There are two states in the signal line, using logic 1 (high) and logic 0 (low) to distinguish respectively. For example, when the transmitter is idle, the data line is in the high logic state. Otherwise when a word is given to the UART for asynchronous transmissions, a bit called the "Start Bit" is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver into synchronization with the clock in the transmitter. These two clocks must be accurate enough to not have the frequency drift by more than 10% during the transmission of the remaining bits in the word. After the Start Bit, the individual data bits of the word are sent, with the Least Significant Bit (LSB) being sent first. Each bit in the transmission is transmitted for exactly the same amount of time as all of the other bits, and the receiver "looks" at the wire at approximately halfway through the period assigned to each bit to determine if the bit is a 1 or a 0. For example, if it takes two seconds to send each bit, the receiver will examine the signal to determine if it is a 1 or a 0 after one second has passed, then it will wait two seconds and then examine the value of the next bit, and so on. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates.

The Parity Bit may be used by the receiver to perform simple error checking. Then at least one Stop Bit is sent by the transmitter. When the receiver has received all of the bits in the data word, it may check for the Parity Bits (both sender and receiver must agree on whether a Parity Bit is to be used), and then the receiver looks for a Stop Bit. If the Stop Bit does not appear when it is supposed to, the UART considers the entire word to be garbled and will report a Framing Error to the host processor when the data word is read. The usual cause of a Framing Error is that the sender and receiver clocks were not running at the same speed, or that the signal was interrupted. Regardless of whether the data was received correctly or not, the UART automatically discards the Start, Parity and Stop bits. If the sender and receiver are configured identically, these bits are not passed to the host. If another word is ready for transmission, the Start Bit for the new word can be sent as soon as the Stop Bit for the previous word has been sent. Because asynchronous data are "self-synchronizing", if there are no data to transmit, the transmission line can be idle.

Example

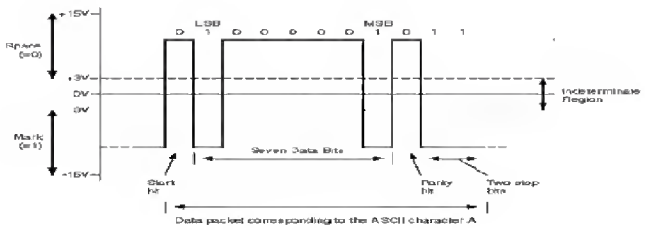

Figure 2

When a word is given to the UART for Asynchronous transmissions, a bit called the "Start Bit" is added to the beginning of each word that is to be transmitted. The Start Bit is used to alert the receiver that a word of data is about to be sent, and to force the clock in the receiver After the Start Bit, the individual bits of the word of data are sent, with the Least Significant Bit (LSB) being sent first into synchronization with the clock in the transmitter. When the entire data word has been sent, the transmitter may add a Parity Bit that the transmitter generates. The Parity Bit may be used by the receiver to perform simple error checking. Then at least one Stop Bit is sent by the transmitter. If incorrectly formatted data is received, the UART may signal a framing error. If another byte is received before the previous one is read, the UART will signal an overrun error.

BLOCK DIAGRAM

Transmitter

We can use any serial transmission unit as source like output from computer, laptops etc. but specifically we select USB port (PC com port) of laptops which provides serial communication protocol, to send data on channel. The load impedance should match with channel impedance. And channel impedance should match with source impedance i e to have efficient data communication, we should have to shift level of data according to channel, so we used level shifter (MAX232). The function of transmit module is to convert the sending 8-bit parallel data into serial data, adds start bit at the head of the data as well as the parity and stop bits at the end of the data.

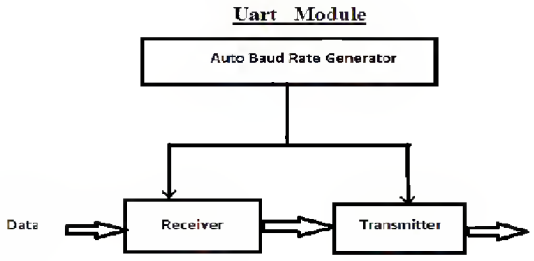

Figure 3

Receiver

During the UART reception, the serial data and the receiving clock are asynchronous, so it is very important to correctly determine the start bit of a frame data. The receiver module receives data from RXD pin. RXD jumps into logic 0 from logic 1 can be regarded as the beginning of a data frame. When the UART receiver module is reset, it has been waiting the RXD level to jump. The start bit is identified by detecting RXD level changes from high to low. In order to avoid the misjudgment of the start bit caused by noise, a start bit error detect function is added in this design, which requires the received low level in RXD at least over 50% of the baud rate to be able to determine the start bit arrives. Since

the receive clock frequency is 16 times the baud rate in the design, the RXD low level lasts at least 8 receiving clock cycles is considered start bit arrives. Once the start bit been identified, from the next bit, begin to count the rising edge of the baud clock, and sample RXD when counting.

Baud Rate Generator

It is actually a frequency divider. The baud rate frequency factor can be calculated according to a given system clock frequency (oscillator clock) and the requested baud rate. The calculated baud rate frequency factor is used as the divider factor. In this design, the frequency clock produced by the baud rate generator is not the baud rate clock, but 16 times the baud rate clock. The purpose is to precisely sample the asynchronous serial data at the receiver. Assume that the system clock is 32MHz, baud rate is 9600bps, and then the output clock frequency of baud rate generator should be 16 * 9600Hz. Therefore the frequency coefficient (M) of the baud rate generator is: M =32MHz/16*9600Hz=208. We are implementing dynamic serial communication and according to our prototype we required the most important block is UART which is implemented on FPGA with the help of VHDL code formula for calculation of baud rate is;

Baud Rate =Clock Frequency/ (Sampling Rate) X (Divisor)

Table 1

Sr No	Baud Rate	Frequency = 50 Mhz	
		Divisor	% Error
1	1200	20833	0.33
2	2400	10416	0.66
3	4800	5208	0.33
4	9600	2604	0.16
5	19200	1302	0.8
6	38400	651	0.04
7	57600	434	0.02

Complete UART Circuit

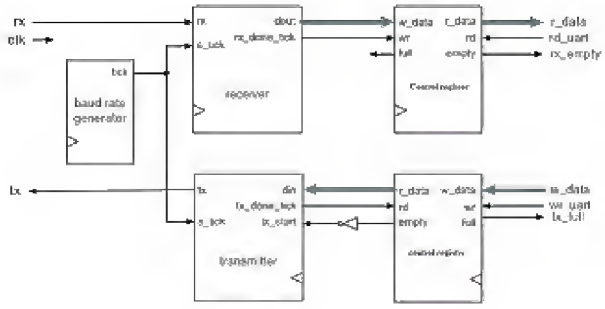

Figure 4

FPGA

FPGA (Field Programmable Gate Array) is using extensively and playing more and more important roles in the designing of digital circuit. Its programmable characteristics make circuit design much more flexible and shorten the time to market. Using FPGAs can also improve the system's integration, reliability and reduce power consumptions. FPGAs are always used to implement simple interface circuit or complex state machines to satisfy different system requirements.

The programming of the FPGA is done using a logic circuit diagram or a source code using a Hardware

Description Language (HDL) to specify how the chip should work. FPGAs have programmable logic components called, logic blocks, and a hierarchy or reconfigurable interconnects which facilitate the ‚wiring' of the blocks together. The programmable logic blocks are called configurable logic blocks and reconfigurable interconnects are called switch boxes.

FPGA Architecture

The architecture comprises Configurable Logic Blocks, Configurable I/O blocks and Programmable Interconnects. It also houses a clock circuitry to drive the clock signals to each logic block. Additional logic resources like ALUs, Decoders and memory may be available. Static Ram and anti-fuses are the two basic types of programmable elements for an FPGA. The number of CLBs and I/Os required can easily be determined from the design but the number of routing tracks is different even within the designs employing the same amount of logic.

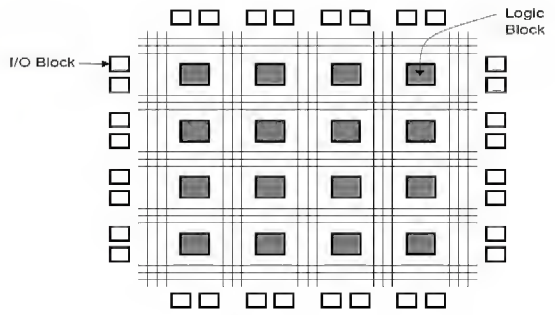

Figure 5

FPGA Design Flow

The flow for the design using FPGA outlines the whole process of device design, and guarantees that none of the steps is overlooked. Thus, it ensures that we have the best chance of getting back a working prototype that will correctly function in the final system to be designed.

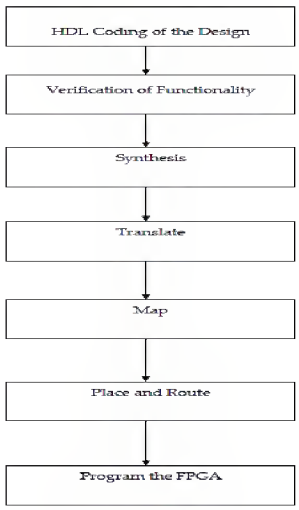

Figure 6

- Initially we have HDL coding of design. Basically it consists of three modules, which design using VHDL language. Hardware behavior of design is explained by VHDL language.

- Design is compile and simulate on model sim software; that is verification of function
- At this step we verify our design using model sim.
- During synthesis; we synthesize our design on particular FPGA platform.
- After synthesis we generate a bit stream file for programming FPGA.

Application

Implementation of UART for serial communication used in following application,

- Synchronization of Telecommunication routers in Telecommunication & Handheld terminals
- Mobile Computing in wide area of network & Factory Automation
- Point-of-Sale terminals also in Data Concentrators. Gaming terminals.
- Cellular data transmission and reception.
- In above all system's to achieve the most secured communication we used UART.

CONCLUSIONS

This paper describes the architecture of UART that support various data word length, parity selection and different baud rates for serial transmission of data. Working principle of this UART has been tested using ISE simulator, which can be implemented on FPGA.

REFERENCES

1. L. K. Hu and Q.CH. Wang IEEE Paper on, "UART-based Reliable Communication and performance Analysis", Computer Engineering, Vol 32 No. 10, May 2006, pp15-21
2. J. Birkner et al, "A very-high-speed field-programmable gate array using metal-to- metal anti fuse programmable elements," Microelectronics Journal, v. 23, pp. 561-568.
3. Amanpreet Kaur, Amandeep Kaur ,International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.2305-2311 "An Approach For Designing A Universal Asynchronous Receiver Transmitter (UART)"
4. Stephen Brown and Jonathan Rose Department of Electrical and Computer Engineering University of Toronto have tutorial on Architecture of FPGAs and CPLDs.
5. "Design of Multi-Channel UART Controller Based On FIFO and FPGA "ISSN: 2278 – 1323 International Journal of Advanced Research in Computer Engineering & Technology Volume 1, Issue 4, June 2012
6. uarttechsupport@exar.com